

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

**Ken Leonard:** ITS Standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems. I'm Ken Leonard, the Director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS standards training program. We are pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience, without the need to travel. After you complete this training, we hope that you will tell your colleagues and customers about the latest ITS standards and encourage them to take advantage of these training modules as well as archived webinars. ITS Standards training is one of the first offerings of our updated Professional Capacity Training Program. Through the PCB program we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at [www.pcb.its.dot.gov](http://www.pcb.its.dot.gov). Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments and thank you again for participating and we hope you find this module helpful.

**Narrator:** Throughout the presentation this activity slide will appear indicating there is a multiple choice pop quiz following this slide. You will use your computer mouse to select your answer. There is only one correct answer. Selecting the submit button will record your answer and the clear button will remove your answer if you wish to select another answer. You will receive instant feedback on your answer choice. Please help us make even more improvements to our training modules by completing the post course feedback form. This is module T203 part one of two, How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two. Your instructor is Manny Insignares. Manny works at Consensus System Technologies as vice president of technology and has over 25 years of professional experience in the transportation system engineering and ITS field. He has significant experience in development and implementation of ITS Standards since 1996, including the first version of the current version three of the traffic management data dictionary standard TMDD. It is now my pleasure to turn it over to your speaker.

**Manny Insignares.** The target audience for this module includes traffic management and engineering staff, maintenance staff, system developers, testing personnel, and private

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

and public sector vendors and manufacturers. And this makes a lot of sense. This is the group that have a vested interest in the system working 100 percent.

Before getting to this course, you may have looked through T101 introduction to ITS Standards Testing, T201 How to Write a Test Plan, and T202 Overview of Test Design Specifications, Test Case Specifications and Test Procedures. This is a diagram showing the curriculum path from T101 through T203 which is this module. We include at the very beginning of the module a list of acronyms so you don't have to go looking through them. So we placed these at the beginning. There are two pages.

This is part one of two. The objectives for this module is to review the role of test cases in the overall testing process. We'll discuss ITS data structures that are used in the NTCIP and center-to-center data dictionary standards such as traffic management data dictionary. We'll find information that's needed to help you develop a test case and we'll explain how to develop a test case. Part two of two will walk through handling standards that are with and without test documentation. We'll develop a requirements to test case traceability matrix. We'll identify types of testing and recognize the purpose of the test logs and test anomaly reports.

Learning objective 1, review the role of test cases within the overall testing process. We're going to review test documentation as it is defined in the IEEE 829 standard, which covers test documentation development. We're going to discuss the test cases and the relationships to test plans, test design, and test procedures. These are other test documentation defined in IEEE standard 829. We'll review some ITS standards testing approaches, the legacy approaches that have been commonly used in ITS, and then review how IEEE standard 829 encompasses many of those or all of those previous legacy approaches and adds some benefits. Do a brief review of module T202, provided a context for testing the lifecycle, introduced you to IEEE standard 829 standard for software and system test documentation. IEEE standard 829 is a guide that goes through the formats of various kinds of test documents. This module will teach you how to use the IEEE approach to develop test cases.

Where do we begin with the test process? Essentially, once you have system requirements, in other words, once you have completed the system requirements phase you're able to continue with design as is shown in the Vee diagram. But you're also able to begin with development of test documentation and specifically test cases.

So what is the purpose of the software testing? These bullets come out of the IEEE 829 standard directly. Just to give you an idea it's to help build quality into the software and

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

system during the lifecycle process, validate that the quality was achieved; to determine whether products from any given lifecycle activity conform with the requirements of that activity. So, for example, we talk about a lifecycle activity it could be unit testing. It could be integration testing. It could be system acceptance testing. So as we're building the system through each phase of the lifecycle we're able to conduct testing. It includes inspection, demonstration, analysis, and testing. The inspection, demonstration, analysis and testing, the IDAT, are used in requirements verification. That is one of the key outputs or key purposes for testing is to verify that requirements have been implemented in the system. And to perform test activities in parallel with development efforts and not just at the conclusion of the development effort. So this testing approach is used throughout the lifecycle, again, after system requirements phase and is used throughout. One of the benefits of being able to conduct testing in an organized way is that to fix problems that you discover in your system, whether it's in the design or it's in the software development, the sooner you discover that problem the less expensive it is to fix. So we want to be able to do testing throughout the lifecycle of the system development, not just at the end.

So what is a test case? A test case specifies the inputs, the outcomes, and the conditions for execution of a test. So if you think of this as a test case is going to write down what are the inputs and outputs for the system that that is what we're testing that's exactly what a test case is going to document for you. Test cases are bundled together in a document called the test case specification and it's part of your overall test documentation. And usually described how it integrates with the other documents in your test plan. Again, this module will focus on the test case specification and test case documentation.

Approaches to preparing project test documentations. We have a couple of NTCIP documents that we're showing here, the NTCIP 8007, which was developed as a guide of what test documentation shall be included in the NTCIP standards. And below we show NTCIP 1204, which is an NTCIP standard for environmental sensor stations and it includes an appendix or an annex that includes test documentation that is formatted as described by NTCIP 8007. The industry is moving forward using the ITS standard 829 approach and we will be discussing some of that rationale momentarily.

The IEEE approach is applicable to all devices. It's applicable for center to center as well as for center-to-field testing. It separates test cases and test procedures, which allows you to invest in developing test procedures which typically are costly, costly to develop, a nice test procedure that can be reviewed across multiple test cases. It includes a test plan and a method to split the testing into what is called test designs. It includes formatting for test reports. Again, the IEEE approach can be broadly applied. So you'll

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

see common formats of test cases, test designs, et cetera, whether you're looking at a center-to-field implementation or a center-to-center implementation.

So what is in IEEE 829? Again it's a guidance document and it specifies the format for various test documentation. And you have a list here of the test plan, the test design specification, test case specification, test procedure specification, test reports, including test logs, test anomaly reports, and test report, that last one is a summary. And it's a format that testing professionals of ITS are familiar with. I first encountered IEEE 829 about seven years ago and in the last seven years I've been encountering many other people in the ITS arena who have looked to IEEE 829 as a way to document their testing process. So now we're at the point here where we're formalizing instruction and providing training on 829 so others can benefit from this documentation.

Let's look at the structure of how these test documents go together. You can see at the top we have a test plan, test design, down through the test reports, and the diagram at the right. The test plan describes the overall approach to testing. It includes information like what the schedule is, what staff are required, if there's any special preparation that needs to happen. In this case we see that there's a master test plan that can be organized by test phase or test activity. We mentioned this when we were looking at the Vee diagram. We have unit tests. Again, we can use the test plan and we test designs and the various documents across the lifecycle there that you see unit testing integration, integration test, system acceptance. And when you're done with implementing your system you can take out the test documentation and it will help you troubleshoot the system. You can do periodic maintenance. So it's a document and an effort that continues throughout life of the system even when it's in operations and maintenance. Test design specification describes which requirements are to be tested and what are their associated test cases. The test case specification, as we'll see in more detail, identifies objectives, inputs, outcomes, and the conditions or execution of a test.

The test procedure specification defines the steps to execute a test. And as you see in the diagram we have a smaller number of test procedures and there is, again, investment that goes into developing the test procedures. It's quite an effort to create a good test procedure and it will be reused with multiple test cases. So this is one approach for reducing the cost of developing test documentation, is to separate test procedures from test cases. After you execute your test or during execution, you will develop reports, such as the test logs, test anomalies reports, and the test report. Some key differences between the two approaches, that is the legacy approach in ITS and the IEEE. IEEE standard approach is applicable to all ITS standards including the center to center and center to field. The IEEE approach separates test cases from test procedures. Previously, such as in the NTCIP standards since they follow the formatting of 8007 test

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

procedures and test cases were combined to one report. The IEEE approach allows you to reuse the test procedures. And the IEEE approach includes a test plan and a way to split the testing into test designs. So that if you use the test documentation throughout the lifecycle, throughout the development life cycle of the system. It also includes test reports.

Activity. Which of the following IEEE standard 829 based component describes data inputs and outputs to be tested? The answer choices: a) test plan, b) test case specification, c) test design specification, d) test procedure specification. Let's review the answers. The correct answer is test case specification. It is the test case specification that focuses on the data input and outputs to be tested. The test plan describes your overall testing approach. The test design specifies your requirements to be tested and which test cases are associated with which requirements. And the test procedure specification outlines the steps to execute a test.

Summary. We reviewed a test documentation structure, how the different test documentation go together in IEEE standard 829. We looked at the relationship of test cases with test plans, test designs, and test procedures. And we contrasted the legacy ITS standards approach such that you would find in the NTCIP standards with that of IEEE standard 829.

Learning objective 2, discuss ITS data structures used in NTCIP and the center-to-center standards such as TMDD, and provide some examples. We're going to look at some of the data structures, look at some examples. The test cases, as shown in bullet two are going to verify that you have correct structure of the data that is exchanged between systems. We'll discuss how a test case verifies the correct value of the data and correct data types.

So let's review an information exchange between ITS centers. The centers exchange information using dialogs, the dialog contains messages that are sent between the two systems. There you see the owner center and the external center. Message A is sent from the external center to the owner center and then returns. The messages are formed with data frames and data elements. So this hierarchical structure is common in ITS so what does the testing verify? It verifies the information that's exchanged. Testing verifies that the correct sequence of information is being exchanged. So in this dialogue we show that message A is sent from an external center to the owner center. The owner center returns message B. And that is a given sequence. That sequence is defined in the ITS standards. So you wouldn't expect to send message A and get back message M, or message P, or any other message. The one that's required to come back is message B. So we want the correct sequence of information exchanges to be tested. We want to

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

verify the correct structure of the information. The ITS standards specify this tree-like structure. We use the tree to sort of describe the structure. At the top level we show the root which is the message. And then below are the branches which are the data frames. And in the leaves we see the data elements. At the root level, again, we see the messages so that is the very top. The data frames, the branch level, are reusable bundles of data elements and other data frames. So just to give you a for instance information such as a time, timestamp that may include the hours and minutes, where let's say the hours and minutes are separate data elements, that data frame is reused quite a bit throughout ITS. You know, one has to do with location so we have these data frames that are reused widely in messages. At the very, very bottom, at the leaf level we contain the actual value of the data. That's where we actually see what the data content is.

So let's look at what the test case is interested in. Again, what are the inputs? In this case, message A, input to the owner center, outcomes or predicted results you know the output, if you will, is message B. And we list an execution condition, the sequence that the owner center responds with message B upon receipt of message A from an external center. And that reads pretty much like many of the requirements that you'll see in the design documents, such as traffic management data dictionary.

So we'll look at an example here of a center-to-center dialog, where we want to issue a link status request message to make a request. And what's returned is the link status information which is the link status message. So that is a specific dialog with a specific correct sequence of messages.

Let's look at the data structure of one of these—the link status request. We have a little diagram here that looks like one of those tree-like structures. And at the root we have the name of the message, in this case, the root is called traffic network information request. And that is a reusable message that we can use to query or to make a request or information about the roadway network in traffic management data dictionary, whether we're looking for route status or link status or information about a particular node on the network, we can reuse this message, the traffic network information request message. To the right you'll see some terms that are outlined either in dash dot outline or bold outline, for example, authentication which is shown in dashed and organization requesting which is a solid line. You can see that there's a specific ordering of these that is specified in the standards. For example, organization requesting always comes after authentication and network information type always comes after organization requesting. So this ordering is important and is one of the things that we test, that this message is correctly structured, this is what we test with the test cases. The dash lines in this example show whether information is optional. For example, authentication is not required. It's optional. While

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

organization requesting is mandatory. It is required to have that information as part of a link status request.

Constraints on content on the data values. The device standards specify correct values in objects. You've heard of the NTCIP objects and it is there where we define the correct value of any particular piece of data. The ITS standards for center to center use an XML format also described in ASN.1 format. And the center-to-field standards define their object definitions in something called a MIB, management information base. Some typical value constraints on the data include the data type. Is it a text or is it a number? For example, if I was looking for the hour or the minutes I would expect only to see numbers zero through nine. I wouldn't expect to see any alphabetic information or any kind of punctuation marks, none of those. So when we constrain a data value, we want to know is it only text is allowed, or only numbers are allowed. Sometimes we have enumerations, which some people refer to as just the standard list of values. So we see that there are a number of places in ITS where we have specific lists with specific values. And the correct value of data is only acceptable or it's only correct if it comes from that list. Other constraints are text length. There are various lengths that are assigned to different texts depending on what the purpose is of that text in the ITS standards. But, again, we have a constraint on what are acceptable data values. There are numerical value ranges, such as 0 to 255 in many of the NTCIP objects. It's also worth noting that in some cases the starting point is a zero when you're counting, and sometimes it's a one.

Here's an example, object NTCIP object from the NTCIP 1204 standard, environmental sensor stations. In this case, we're looking at wind sensor situation. You can see there, 5.6.10.10. That's the paragraph in the standard. And we'll see that only number values are valid in the values one through twelve. So here's the constraint that it is a number value. You'll see that it is an integer. And we constrain its value here with a valid value list. You see things like "other" and "unknown" and "calm, light breeze is four," et cetera. Only the number is transported, but we see a mapping from that number to some text. And there are only twelve. So if we were developing a test case we would test that the value that's being sent over is, in fact, between one and twelve and it's not some other value. Any other value would be incorrect.

Here are some places where you can find the constraints for these different data. For center-to-field NTCIP devices you can look in something called the management information base. You can also look in the section where the ASN.1 objects are defined. For the center-to-center standards such as TMDD you can look in Volume II which is the design and you will see a value specification written in XML format and another value specification representing the same constraints in ASN.1 format. And you will also find the data structure that's acceptable. You'll see messages listed out, what all the data

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

frames are, the data elements, and the relationships, which would let you understand as we've seen before the tree-like structure that makes up the messages or makes up some of the more complex objects.

So, again, what is the purpose of the test case? It's to verify the requirements related to the information that's exchanged between two systems. So we want to verify the sequence of information that's exchanged, that's the dialogs showing inputs and outputs. We want to verify the structure of the information is correct, again, the standards define the order of the messages that data frames, how the data elements are stacked. We want to verify the content of the data so we looked at a couple of examples of the valid value rules, which included things like the values ranges, valid value lists, text length, et cetera.

We can relate test cases to requirements. This is typically done in the requirement, the test case traceability matrix. We have an example here in the bottom. This comes out of NTCIP 1204. And on the very left hand side it says requirement. Right below it it says ID. 3.5.1.1.2 at the very top row and its title is retrieve compressed station meta data. Now, on the right hand side is the corresponding test case. In this case, the test case ID is the letter C.2.3.1.2. And the title retrieve compressed station meta data. So in both cases there's a paragraph ID that is used. In the case of the test cases because the test cases and test documentation for NTCIP standards is included in annex C of the standard the paragraph begins with the letter C. As we'll see shortly the ID for test case would pretty much be anything you want it to be as long as it's unique inside the test case specification.

Activity. Which of the following defines the structure and data content of inputs and outputs? Answer choices: a) data dictionary standard, for example, the NTCIP 1204, or traffic management data dictionary, b) the protocol requirements list, c) the requirements to test case traceability matrix, or d) all of the above. The correct answer is the data dictionary. The data dictionary specifies the structure of the data and the constraint of values for data content. The PRL traces requirements to needs. It allows you to specify optional requirements that you want to be mandatory in your project. So the PRL helps you specify, looking from user needs, what are the requirements for your project. The requirements to test case traceability matrix traces test cases to requirements that the test case will verify. And only a is correct.

So in learning objective 2, we review the concept of data structure and how ITS information are structured in the NTCIP standards, TMDD showed some examples. We discussed how test cases verify that you have correct structure of data. We discussed how test cases verified correct value of data. What are the correct value ranges? What

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

are the correct rules for the data? And the data types, and make sure that conform what is specified in the standards, the data dictionary standards.

Learning objective 3. Find information needed for a test case. We review what information is needed. What are your user needs for your project? What are the relevant requirements? What is the relevant design, dialogs, data elements? What are the valid values? Where do we find this content so we can develop a test case when we're dealing with the center-to-center standards, such as TMDD? Where do we find the content we need for a test case when we're looking at the center-to-field standards, such as the NTCIP standards? For the center-to-center standards, we can begin by looking at your needs. These are identified in the needs to requirements traceability matrix, NRTM. Then we'll see how we trace down from the needs to the requirements, down to the design to look up the dialogs, which define the inputs and outputs. Here's an example of a project level NRTM. We call it a project level NRTM because we've filled in a column here to specify whether certain optional elements are required for your project. In this case, I'm looking at user need, has an ID 2.5.2.2, travel time data for roads, and then the requirement that satisfies that need. You can see we have dialogs, you have the request to message, response message, and the error report message described. And on the very last, you simply mark yes or no, whether this dialog and these different elements of the request message and response message are required for your project. We look down at response message, about half way down. There's an element called link name, which is optional, but for the project, you can see the O there is optional, but for your project, we've marked it as yes, it's required. So that becomes a mandatory requirement for your project. So the top part of this shows, again, the NRTM like we did before, like we walked through before in the previous slide. And at the bottom, we're looking at, what are the requirements? Excuse me. We used the RTM at the bottom, the requirements traceability matrix, to trace into the design. So the bottom part shows a trace between the requirement, in this case, 3.5.3.3.2.1 called send link status information upon request. It's a dialog. We'll see that it traces to TMDD Volume II dialog link status request. And here it is in volume 2, it shows where the paragraph is where you can read about that. At the top we show that, no we are not using these optional elements, so when we come down here and trace in the RTM, they're shown as crossed out, so they're not required in your project. In a similar way, we can use the protocol requirements list, the PRL, for the NTCIP standards, for the center-to-field standards. In this case, we show a section of the PRL and the NTCIP 1203 dynamic message sign standard. We then go to the requirements traceability matrix to look up the design and look at what are the inputs and outputs? Again, we show on the left the requirement. On the right hand side, dialog and the object definitions.

Activity. Which of the following will provide information on project needs for a center-to-center project? Sources of information include, these are your answer choices, a) needs

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

to requirements traceability matrix, b) requirements to test case traceability matrix, c) requirements traceability matrix, or d) the design, dialogs, data elements, valid value range definitions? Answer, needs to requirements traceability matrix identifies your project needs for center-to-center project. The requirements to test case traceability matrix traces test cases to requirements the test case verifies. The requirements traceability matrix traces requirements to the design, and the project design does not trace to project needs, at least not directly and not in the ITS standards. In learning objective 3, we looked at what information we need to develop a test case and we showed some examples with documentation, information you'll find in the center-to-center standards, and the kind of information you would find in the center-to-field standards, such as the NTCIP environmental sensor station.

Learning objective 4 will explain test case development. We'll begin with an outline of a test case, provide a suggested template. We'll walk through what is the required content. We will build upon what we went through in the learning objective 3, the information we need to be able to fill in the test case template. We'll discuss positive and negative testing and additional test case requirements. Here we have an outline test case. On the left hand side we see these bullets that come right out of our IEEE 829. We show the test case identifier, test case objective, inputs, outcomes, environmental needs, special procedural requirements, and intercase dependencies. On the right hand side we have a little template that pretty much puts on the little table. On the left hand side are the required elements. As we walk through, they match the bullets. And on the right hand side we have a blank one, blank information which we will fill in shortly. Let's look at the first item, the test case identifier. Each test case requires a unique identifier to distinguish it from all other test cases. So again it can be whatever you want it to be but it should be unique. In this case, we're showing you the final product of what the test case looks like. You can see at the top, it's identified as TC001. Second item required is a test case objective. We outlined three more bullets here to give you an idea of what should be in an objective. The purpose. The objective should identify what is the purpose of the test case. Focus. If there is a special focus for a particular test case then we would define that as well. And the priority. Is this a test case that is, in terms of timing, should be done ahead of others? Again, it doesn't tell you specifically which test cases, but there may be some group of test cases that you want to do first. The focus. To identify whether the test case is testing a specific dialog, whether it's testing again, what is the correct sequence of messages that are being tested by the test case. Whether the test case is testing for the correct structure and content of data. It could be doing negative testing, which means it's trying to force an error. In intercase dependencies, there's a section just to define the actual test case identifiers, but we can also give a verbal description in the objective of what are some dependencies among the test cases. One example is, if I do a subscription publication, I want to subscribe for updates and then those updates will come back to me as they occur, those are two different things. Two things have to be tested there. The subscription first, in that it be done correctly before I can test any of the

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

publications that come afterwards, the updates. So that is a subtle intercase dependency that I can put in words and make part of my test case objective. The priority, again, identifies a relative importance of test cases, and there are, again, the priority is dependent mostly on how your project is being planned out. Some examples include that you may be installing in your ITS project certain devices ahead of others. Maybe you need to put in the CCTV cameras first and then you'll be putting in the dynamic message signs and perhaps other field equipment. So in this case, you would give CCTV test cases a priority. You may want to specify that you want to do inventory. You might want to query the inventory from a center, say the inventory dynamic message signs or the inventory of CCTV first and do all those, and then do the status. What is the status of the dynamic message sign? What is the message that's on it? What is the status of the CCTV camera? Is it online? What direction is it facing, et cetera? You may specify that you want to do all your request response dialogs first before you do subscription publication. Or you may specify that you want to do all your positive test cases, in other words, all the ones where you have proper requests being made, with the expectation of a proper response. You're not trying to force an error, which is what we do in negative test cases. So these are some examples of how your project may want to prioritize which test cases get done, and so you can organize them in your test case specification.

Test case inputs. Specify each input that's required to execute a test case. Some inputs will be specified by value. What are the tolerances? Others will be included in tables or transaction files, but we will specify the input and the timing of that input that's required for the test case. Here's an example of a test case input specification. We looked earlier at the link status request message. So this is the test case input specification for the link status request, and it shows that this is going to be used with a positive test case. In other words, it's going to be a proper request, followed by a proper response. We show below, just like we did in the graphical example prior, a traffic network information request message, it's a message. We list out the data frames and when we get to the data element, such as the organization ID or the data element organization name, we then show its value constraints, which we call the value domain. In this case, we have a little bit of ASN.1 formatted specification here. It shows a value specification for organization ID. It is a string and its size is one to 32. So it's a piece of text that can be as short as one character and as long as 32. So it cannot be zero, for example, it cannot be 33, or it would create an error. Here we have network information type. Again, another data element. In this case, we simply list what is the valid value list. Since I'm going to request the link status, then I need to have the number 4 or the text link status included in my request.

Test case outcomes. Here we specify all the outputs and expected behavior, such as response time for the various insider test case here. And again, we provide representative values for the required output. So when we send our link status request,

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

expect to get link status information back. This would be the proper response. We're doing the positive test case, so we expect to get good information back. Similar to the test input specification, we show the data elements here, the data concepts, whether it's a message, a data frame, data element, and a value constraint. Organization ID and organization name have the same constraints as shown before, one to 32, they're text. And we have another one here, link status could be one, two, three, or four. We may want to show what the travel time is on that link. In this case, we show that it's an integer between zero and 65,535 and that the units are in seconds.

So here we have the completed, filled in test case specification where we show unique ID and a title this is a link status request response, dialog verification for a positive test case. We want to verify the system interface implements, the requirements for link status request response, that the contents of the link status request message is correct, contents of the link status information message are correct. We describe the inputs, in this case, we reference the test case input specification, which we reviewed a few slides back. We also indicate that we have an example, link status request dot xml, which we can use as an example for what this correct input will look like. Same for the outcome. We point to the test case output specification. We indicate that all data are returned and verified as correct.

Environmental needs. There are no additional needs outside of those specified in the test plan. Add another bit of information here, who the tester or the reviewer was, which is an optional element in the test case specification. And then we have the last two items, which are required, special procedure requirements. Special procedural requirements, there are none, but if there were any, a special procedure or a requirement or something that is outside of what is specified in your test procedure, then you would include it in this box here. And then intercase dependencies. There are none but if there were, you would identify what the number of the test case is that needs to be done before you use this one.

Positive test cases. So positive test case inputs and outputs include data values being within a correct range. Want to show that the data structure is correct, as specified in the standard, that all mandatory data values are included, so that includes both the ones that the ITS standards say are mandatory, but also if your project defines or makes certain optional elements of the standard mandatory, then you have to test for those. So you want to test that all mandatory data values are present.

Here's a simple example, shows a bit of xml for traffic network information request message all packaged up. We've got the authentication here, we've got the user ID, password, organization requesting, organization ID, and in bold, we show what the data

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

value is, what the data content is. The negative test case include data values that are not within the range of values. Could be that your text length is too big or invalid. There are a number of value specifications where the value, the text length must be, for example, one to 32. We saw an example of that earlier. Well, if it's zero, that's not correct. We have to find that, so that's an error. In this case, with a negative test case, we're trying to force an error to make sure that the system responds properly, in other words, it handles errors in a correct way. You may have data that's not correctly structured or you may have missing mandatory data elements.

So let's look at some examples of those. Here we have a negative test case example and it actually creates, I show three kinds of errors here. The errors are listed over here on this little box. Invalid username and password. So in this case, we have the correct user ID, but the password is not correct. We have missing mandatory element, organization ID is required. So I show it here in this xml. You can see where it says organization ID. This shows that it's a comment, a little exclamation point with a dash dash, so it's really not there. This would be an error. Or we may have extra elements defined and they are not allowed. So in this case, I added something from, those of you familiar with accounting, an accounting term here called the appreciation method and sum of the year's digits. So that would be something extra that you wouldn't typically find in something in ITS. So those are a message that has three errors, and you need to define test cases. We have three error paths here and you need to define three separate cases, test cases to be able to trap these three kinds of errors. Missing elements and incorrect data. Here's an example of a missing element. On the left hand side, we see the correct message and we show before that Org 001 is a correct organization ID. But it doesn't show up on the right hand side, so we have a missing mandatory element. The organization ID is not there, so that's an incorrect request message. My expectation would be, if I sent an incorrect message to a center, requesting link status or link inventory, as is shown here, that I should get some kind of an error back.

Here's an example of incorrect data structure, a fairly simple example, but you can see that on the left hand side, the user ID and the password are correct. They're shown in blue there, user ID comes first, and then password. On the right hand side, password comes first, then the user ID. That is not correct. So you have to have the sequence of these data elements just right.

Test case environmental needs. These describe your test environment that's needed for your setup, your execution, and what results will be recorded. Ideally, this information is included in your test plan. So what we have in the test case is an opportunity. We have a box for it, as we showed in the filled out example. That if you have some additional requirement that's not described in the test plan or it differs from what's in the test plan,

## T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two

then you should write that down. What are the test case specific environmental needs? Otherwise, we simply say that there are no specific or new environmental needs other than what's described in the test plan. This section of the test case may just reference the test plan, include a paragraph, where in the test plan do you describe environmental needs? But if there is a case where you need to do something special or something that is different, especially if there's something different than what's in the test plan, then we need to include that in the test case so that we know that something—some sort of a special case being tested here.

Test case special procedural requirements, similar kind of a thing. The procedures, the steps that you take to execute a test are in your test procedure specification. Again, if your test case is going to require something special or something that is not included in the test procedure, or it differs, contradicts what's in the test procedure, then there's an opportunity in the test case to define what that is. So it would also be a place to describe any pre- and post-conditions for your test case execution. The intercase dependencies for your test cases, again, it's list the identifiers of the test cases that must be executed prior to this test case's. Summarizes the nature of these dependencies and the example we've used is, the subscription publication. There are other dialogs where it would be, depending on how this is set up, that if you're testing an NTCIP dialog and you're requesting several objects and you're requesting them in a specific order, and if you are creating a test case for each of these, then in your test case, you will describe which are the previous test cases that you need to be conducting. You put them in a section called test case intercase dependencies. All right, activity. Which of the following is part of the IEEE 829 test case specification? Answer choices, description and valid values of inputs and outputs? Project sponsor? Steps to conduct a test? Test, pass/fail? And the answer, the description of the valid values of inputs and outputs. Yeah, the test case includes specification of the inputs, outputs, and their proper values. The project sponsor is not a formal part of a test case. Steps to conduct a test is contained in a test procedure. And the pass/fail information is typically contained in a test procedure.

Summary of learning objective 4, we reviewed outline of the test case and the suggested template. Started with a blank template and filled it in. We discussed where to find information to fill in your test case template, both from the center-to-center and the center-to-field standards. We discussed positive and negative testing and we reviewed some additional test case requirements.

So here we are at the end of the module. What did we learn? What were the things? Let's do a review. Role of test cases in relation to other test documents, including your test plan, test designs, test procedures, test reports. What the purpose of the test case specification is, which is to document the inputs, expected outcomes and execution

## **T203 How to Develop Test Cases for an ITS Standards-Based Test Plan, Part One of Two**

conditions for a test. The outline for a test case specification is identified in IEEE standard 829. Then we list A through G, the required elements as described in 829. We learned that ITS data dictionary standards constrain the structure of data and the content of data, of information exchanges between systems. We walked through an example. Test case to learn how to develop one. Here's some of the resources, some places you can go to, to get more information. These are also in your student supplement. There is a part 2 to this test case, to this module on test cases where we will learn how to handle standards that are with and without test documentation. We'll develop the requirements to test case traceability matrix, and you'll identify types of testing, and recognize the purpose of test logs and test anomaly reports. At this point, I want to remind you to please complete the feedback form which is presented to you at the end here. And with that—I'm complete—I'm done.

##### End of T203 How to Develop Test Cases for an ITS Standard Based Test Plan,  
Part One of Two #####